

ROUTING AND DESIGN IN K-SHARED NETWORKS

Cross Reference to Related Application

This application relates to the U.S. patent application identified as Serial No. 10/392,574
5 (attorney docket no. Alicherry 1-1-1-1-16), entitled "Low Latency Shared Data Path Allocation,"
filed March 20, 2003 and commonly assigned, the disclosure of which is incorporated by
reference herein.

Field of the Invention

The present invention relates to network routing and design techniques and, more
10 particularly, to routing and design techniques for use in accordance with networks such as K-
shared networks.

Background of the Invention

The last few years have witnessed the introduction of optical (and electrical) mesh
15 networks as an alternative to synchronous optical network-based (SONET-based) ring networks.
One of the key benefits of mesh networks is the improved bandwidth utilization coming from
path diversity and shared restoration. Unlike the traditional 1+1 protection technique which
reserves 50 percent of the bandwidth for backup paths, shared restoration allows multiple
demands to share backup channels and hence reserves less capacity.

20 However, shared restoration comes at the cost of increased restoration times. Industry
standards demand that circuits at the transmission layer be restored within, for example, 50
milliseconds in order to support voice and other SONET traffic. This is easy to achieve with 1+1
protection because traffic is simply switched to a pre-setup backup path on failure and operations
resume almost instantaneously. Under shared restoration, since a backup channel may belong
25 to multiple paths, it is not possible to set up the backup paths a priori. Instead, the path is
precomputed, but set up after a failure. Set up involves signaling, acknowledgments, and cross
connect configurations. This typically takes much longer than the required 50 milliseconds,
making the technique non-viable for many applications.

Motivated by these and other deficiencies, a fast shared restoration technique was proposed in the U.S. patent application identified as Serial No. 10/392,574 (attorney docket no. Alicherry 1-1-1-16), entitled "Low Latency Shared Data Path Allocation," filed March 20, 2003 and commonly assigned; and also in C. Phadke et al., "FASTeR: Shared restoration without signaling," *OFC*, March 2003, the disclosures of which are both incorporated by reference herein. In accordance with the fast shared restoration technique, if the degrees of a shared channel (i.e., the number of shared channels incident on that channel at its ends) are restricted to a finite number (K), then one can in fact pre-establish the backup paths and restore them almost as fast as 1+1 protection. Networks implementing this fast shared restoration technique are referred to as K -shared networks. However, traditional routing and design algorithms may not always work efficiently with networks implementing such a fast shared restoration technique.

Thus, a need exists for routing and design methodologies for use with networks implementing such a fast shared restoration technique and the like.

Summary of the Invention

The present invention provides network routing and design techniques.

In one aspect of the invention, a technique for determining a route for a demand in a network, wherein the network comprises primary paths and secondary paths, and at least two secondary paths may share a given link, comprises the following steps/operations. First, a graph representing the network is transformed. Edges of the graph represent channels associated with paths and nodes of the graph represent nodes of the network. The transformation is performed such that costs associated with the edges reflect costs of using channels in secondary paths. Then, the shortest path between nodes corresponding to the demand is found in the transformed graph. The shortest path represents the least-cost path in the network over which the demand may be routed.

In another aspect of the invention, when the above route determination steps/operations result in a path with at least one loop, an alternative routing process is executed so as to determine a loopless path for the demand.

In yet another aspect of the invention, integer linear program (ILP) formulation design techniques are provided. For instance, a technique for designing a K-shared network based on a set of one or more demands comprises the following steps/operations. Candidate primary paths and candidate secondary paths are computed based on the set of one or more demands. At least
5 two candidate secondary paths may share a given channel and the number of shared channels incident on another channel is a finite number K. An integer linear program formulation is applied to the computed candidate primary paths and candidate secondary paths. The integer linear program formulation applied to the computed candidate primary paths and candidate secondary paths is then solved so as to generate a K-shared network design.

10 While the routing and design techniques of the present invention may be employed in a variety of network types, they are particularly well-suited for implementation in accordance with K-shared networks.

These and other objects, features and advantages of the present invention will become apparent from the following detailed description of illustrative embodiments thereof, which is
15 to be read in connection with the accompanying drawings.

Brief Description of the Drawings

FIGs. 1A and 1B are diagrams illustrating a K-sharing technique for fast restoration in a network;

FIG. 2A is a flow diagram illustrating a routing methodology according to an embodiment
20 of the invention;

FIG. 2B is a flow diagram illustrating a transformation process for use in a routing methodology according to an embodiment of the invention;

FIGs. 3A and 3B are diagrams illustrating a transformation process for use in a routing methodology according to an embodiment of the invention;

25 FIG. 4 is a diagram illustrating a network with a looped path;

FIG. 5A is a flow diagram illustrating a routing methodology according to another embodiment of the invention;

FIG. 5B is a flow diagram illustrating a process for computing a corresponding cheapest loopless K-shared path for use in a routing methodology according to an embodiment of the invention;

FIG. 6 is a flow diagram illustrating a K-shared network design methodology according to an embodiment of the present invention; and

FIG. 7 is a block diagram illustrating a generalized hardware architecture of a computer system suitable for implementing an automated K-shared routing and design system according to an embodiment of the present invention.

10 **Detailed Description of Preferred Embodiments**

The following description will illustrate the invention in the context of an exemplary optical network. It should be understood, however, that the invention is not necessarily limited to use with any particular type of network. The invention is instead more generally applicable to any environment in which it is desirable to perform efficient demand routing and network design. Thus, by way of example only, the techniques of the invention may also be applied to circuit-switched networks such as SONET and multi-protocol label switching (MPLS) networks. Also, it is to be further appreciated that the present invention may be used to compute paths for optical burst switching.

As illustratively used herein, a “channel” may be a wavelength in an optical network, a “link” may couple two nodes and may include one or more channels, and a “path” may include one or more links.

As will be evident from the illustrative embodiments described herein below, the present invention provides several advantageous contributions. For example, the invention provides an optimal routing methodology for K-shared networks that minimizes the resources used for sharing. Also, the invention provides an efficient heuristic routing methodology that proves that loopless routing in a K-shared network is NP-Hard. The invention also provides an integer linear program (ILP) formulation of the network design problem that minimizes the capacity required. Experimental results show that K-shared networks offer nearly as much bandwidth savings as unrestricted sharing, even for small values of K, e.g., where K is four or less.

For the sake of convenience, the remainder of the detailed description will be divided into the following sections: (I) K-shared Networks; (II) Routing in K-shared Networks; (III) Design of K-shared Networks; and (IV) Illustrative Hardware Implementation.

I. K-shared Networks

5 In this section, the K-sharing technique for fast restoration is described. It is assumed that the network comprises bi-directional optical links interconnected by optical cross-connects (OXC) in an arbitrary mesh topology. However, the description applies equally well to electrical mesh networks. Each link is assumed to have at most m wavelength channels. Traffic comprises point-to-point unit wavelength demands which are to be routed over the channels. A path is set
 10 up by configuring the cross-connects to connect an incoming channel to the next channel in the path. In the traditional path restoration techniques, each demand has a dedicated primary path and a precomputed backup (secondary) path, and the backup paths of multiple demands may have common channels. Clearly, only one of these demands with common channels can be reverted to the backup path on a failure. In link restoration, there will be a backup path per primary link,
 15 but K-sharing also works for that case. On a node or channel failure in the primary path of a demand, its backup path is established by sending signals to all the cross-connects along the path. The traditional protocol for this involves the following steps: send signals to the OXCs along the backup path, process signals and setup cross-connects, and wait for acknowledgments (ACKs). Given the size of the network and the non-trivial latencies in OXCs, this can take considerable
 20 time.

The K-sharing technique was designed to eliminate signaling and cross-connect latencies from the critical path of restoration. It works as follows. Even before any failure occurs, all the backup paths are set up using a specialized *Select-Multicast* feature of the OXCs. This is explained through the scenario illustrated in the context of FIGs. 1A and 1B where two demands
 25 $D1$ and $D2$ overlap in their bidirectional backup paths. The backup channels $l1$, $l2$ from the two backup paths are incident on node $N6$ and merge onto a single backup channel L . This node would be configured to behave as follows: (a) the node sends the signal from the first channel among $l1$ and $l2$ with data on it on to L , while blocking the remaining channel (*Select*); and (b) the light coming on L into $N6$ is always broadcast onto $l1$ and $l2$ (*Multicast*). On a failure in a

primary path, say of $D1$, the end-nodes of $D1$ ($N1$, $N3$) transmit data right away on the backup path. If this was the first failure among $D1$ and $D2$, the signal will reach the intended destination after successive multicasts, while other demands are prevented from using the backup channels through blocking. If this happens after $D2$ had failed and been restored, the light on $I1$ would be
 5 blocked at $N6$ and will never reach the destination, as expected for any shared restoration technique.

Essentially, blocking replaces the more complex cross-connect setups and multi-casting eliminates the need to process address headers in the internal nodes. The latter is useful for reducing delays as well as in all-optical networks where the intermediate nodes do not have
 10 capacity to interpret data. Since no cross-connects are setup and signals processed, the time to restore is roughly equal to the backup path latency.

In practice, it may not be easy to build an OXC with a high *Select-Multicast* degree (or, simply, degree where no confusion arises), i.e., the maximum number of channels connected to a single channel. The maximum degree allowed in a given network is in fact the K in K -sharing.
 15 Many current OXCs provide functionality which can be easily adapted to implement a *Select-Multicast* feature. However, the degree supported for those applications is typically just two and extending this degree to four might mean further doubling the number of mirrors in a micro electrical-mechanical system (MEMS) architecture. On the other hand, a small degree has a negative impact on sharing because certain backup path configurations would be disallowed.
 20 Note that, since each backup channel may already belong to multiple demands, this does not restrict the number of demands sharing a backup channel to K .

II. Routing in K -shared Networks

When a new demand arrives, its primary and backup paths need to be computed. The methodologies for traditional shared restoration are well known, see, e.g., J.W. Suurbaale,
 25 “Disjoint paths in a network,” *Networks*, vol. 4, June 1974, the disclosure of which is incorporated by reference herein. However, the backup paths computed by these traditional methodologies may violate the maximum degree constraint in a K -shared network. In this section, an optimal least-cost routing methodology is illustrated for K -shared networks, which minimizes the number of channels used for backup at least-cost.

Let $G = (V, E)$ be the network, where V is the set of OXC nodes and E is the set of wavelength division multiplexing (WDM) links connecting the nodes. Each link $l = (u, v)$ carries m channels l^1, l^2, \dots, l^m . For a channel l^i incident on a node u , its degree at u is defined to be the number of backup channels connected to it at u .

5 Now, consider routing a demand d between nodes s (source node) and t (destination node). A channel l^i can be in one of the following states at any given time in regard to d and is assigned a cost $c(l_i)$ as stated:

- 1) The channel is part of the primary path of a demand and hence not available.
- 2) The channel is a backup channel for a demand sharing a risk group with d (e.g., having
10 common links in primary paths) and hence not available.

3) The channel is a backup channel, but not for any demand sharing a risk group with d . The channel can be used for d , but the next channel in the path should either be one of the channels it is already connected to, or both channels should have degrees less than K at their common node. A cost of zero is attributed to using that channel in this case.

- 15 4) The channel is not used by any path. It can be used for the backup without any restrictions, but at a cost of one channel.

Referring now to FIG. 2A, a flow diagram illustrates a routing methodology according to an embodiment of the invention. Routing methodology 200 comprises the following steps. In step 202, transform the graph G into $G' = (V', E')$ such that edge costs capture the cost of using
20 the channel in a backup path. Then, in step 204, find the shortest path between the nodes corresponding to s and t in G' . This will directly yield the cheapest path in the network.

Referring now to FIG. 2B, a flow diagram illustrates a transformation process (e.g., step 202) for use in a routing methodology according to an embodiment of the invention. Transformation process 250 involves the following operations:

- 25 1) Transform channels into nodes (step 252): Consider any link $l = (u, v)$ and channel l^i in it. Create two nodes $u(l, i)$ and $v(l, i)$ and an edge between them. The cost of this edge is the cost $c(l_i)$ as stated above.

- 2) Capture channel connectivity (step 254): Consider any pair of channels l^i and m^j incident on a node u . If these channels are already connected at u , then add a zero-cost edge
30 between $u(l, i)$ and $u(m, j)$. Otherwise, if both channels have degrees less than K at u , then

connect them with an edge costing ϵ ($0 < \epsilon \ll 1$). This cost is to prefer existing connections over new ones, because a new connection would increase the degree of the channel.

- 3) Include source, destination (step 256): Create nodes s' and t' corresponding to s , t . Consider channel l_i in link $l = (s, x)$ incident on s . If its degree at s is less than K , connect s' and $s(l, i)$ at zero cost. Repeat with t .

The complexity of this least-cost methodology is $O(em^2 D \log(em))$, where e is the number of links in G and D is the average nodal degree.

The transformation may be illustrated as follows.

- FIG. 3A shows node A connected to four physical nodes B , C , D , and E . Let $K = 3$ and number of channels $m = 2$. The first channels of all of the links are connected to each other through the switch of node A . The second channel of link AB is connected to the second channel of AC , and the second channel of AD is connected to second channel of AE .

- FIG. 3B shows the graph transformation. The nodes corresponding to the channels already connected in the network are joined with links (solid lines) of zero cost (e.g., step 252). The nodes corresponding to the second channel that are not connected in the switch are connected using links (dashed lines) of cost ϵ since their degrees (2) are less than K (e.g., step 254).

- While the above methodology computes a least-cost backup path, it is possible that the resulting path may visit the same node more than once. FIG. 4 shows an example for $K = 2$ for a path from A to E . Only shared channels are shown here. Since the channel in link AB is already connected to two channels BF and BC , it cannot be connected to another channel BE . So the routing methodology takes the path $A - B - C - D - B$. Such paths are often not acceptable in practice for several reasons. First, they are unduly long and may introduce excessive latencies for delay-sensitive traffic. Second, all-optical networks are designed such that a signal is transmitted with sufficient power between any two nodes only as long as the paths have a finite maximum length. Since loops violate this property, they are not permissible.

- In the above example, an alternative solution would have been to use another channel in the link AB for the backup path and connect it to BE to get the path $A-B-E$. However, the general problem of finding such loopless paths is non-trivial. That is, it may be proven that, given a K -shared network, computing the shortest path without visiting the same node twice is NP-Hard.

Next, an efficient heuristic loopless routing methodology for this problem is provided, which should be invoked when the least-cost routing methodology results in a path with loops.

Referring now to FIG. 5A, a flow diagram illustrates a routing methodology according to another embodiment of the invention. Loopless routing methodology 500 first finds r shortest paths between s and t in the graph G using the well-known Lawler's algorithm, in step 502. Lawler's algorithm is described in E.L. Lawler, "A Procedure for Computing the K Best Solutions to Discrete Optimization Problems and Its Application to the Shortest Path Problem," *Management Science*, vol. 18, no. 7, pp. 401-405, March 1972, the disclosure of which is incorporated by reference herein. The complexity of this operation is $O(rn^3)$, where n is the number of nodes.

For each of these paths, in step 504, the methodology then computes a corresponding cheapest loopless K-shared path as described below. Finally, in step 506, the methodology selects the cheapest among these r K-shared paths as the backup path.

Referring now to FIG. 5B, a flow diagram illustrates a process for computing a corresponding cheapest loopless K-shared path (e.g., step 504) for use in a routing methodology according to an embodiment of the invention. Computation process 550 involves the following operations in order to find the cheapest K-shared path corresponding to a given path.

In step 552, let the path be $P = u_1 - u_2 - \dots - u_r$. In step 554, build a directed graph S containing only the nodes and links from P . The direction of a link is in the direction of the path. Then, in step 556, transform S using the transformation used in the least-cost routing methodology described above, with u_1 and u_r as the source, destination nodes. Also, for the purpose of the transformation, the degrees of the channels are taken from G rather than S . Then, the shortest path from u_1' to u_r' in the transformed graph is the cheapest K-shared path without loops corresponding to P . A proof of correctness may come from the fact that links are directed in the direction of the path P .

III. Design of K-shared Networks

A goal of the network design problem is to assign the minimum capacity (total number of channels) to the links in order to carry a given set of demands. Design methodologies for

traditional shared restoration do not work for this case because they may exceed the degree K at some nodes.

In this section, an integer linear program based methodology is provided for designing K -shared networks. Referring now to FIG. 6, a flow diagram illustrates a K -shared network design methodology according to an embodiment of the present invention. Design methodology 600 begins in step 602 by computing a finite number of candidate primary and backup paths for the demands. Doing this outside of the linear program reduces the number of variables and prunes unnecessary paths. Let $(P_1^d, B_1^d), (P_2^d, B_2^d), \dots, (P_{n_d}^d, B_{n_d}^d)$ be the pairs of paths for demand d . Next, in step 604, the following formulation is applied to design the network.

10 The variables used in the linear program are:

$p_{i,d}$ 1 if the path P_i^d is used for d (else, 0)

$b_{i,d,l,\lambda}$ 1 if the path B_i^d uses channel λ on link l

$u_{l,\lambda}$ 1 if link l uses channel λ for backup

x_l Total number of channels used on link l

15 $c_{l_1,\lambda_1,l_2,\lambda_2}$ 1 if channel λ_1 of link l_1 is connected to channel λ_2 of link l_2 .

The goal is to minimize the total number of channels used $(\sum_l x_l)$ subject to the constraint that $x_l \leq m$, for all l , where m is the number of channels available on a link.

The channels on a link can be used either for primary or for backup:

$$x_l = \sum_{i,d:l \in P_i^d} p_{i,d} + \sum_{\lambda} u_{l,\lambda}, \text{ for all } l$$

20 The various constraints may be as follows:

$$\sum_i p_{i,d} = 1, \quad \forall d \quad (1)$$

$$\sum_{\lambda} b_{i,d,l,\lambda} = p_{i,d}, \quad \forall (i,d) \text{ and } l \in B_i^d \quad (2)$$

$$B_{i,d,l,\lambda} \leq u_{l,\lambda}, \quad \forall l,\lambda, (i,d): l \in B_i^d \quad (3)$$

$$B_{i_1,d_1,l,\lambda} + B_{i_2,d_2,l,\lambda} \leq 1, \quad \forall l,\lambda, i_1,d_1,i_2,d_2: d_1 \neq d_2$$

and paths $P_{i_1}^{d_1}$ and $P_{i_2}^{d_2}$
has a common link

(4)

$$B_{i,d,l_1,\lambda_1} + B_{i,d,l_2,\lambda_2} \leq 1 + c_{l_1,\lambda_1,l_2,\lambda_2}, \quad \forall l_1,\lambda_1,l_2,\lambda_2,i,d$$

: $l_1,l_2 \in B_i^d, (l_1,l_2) \text{ adjacent}$

(5)

$$c_{l_1,\lambda_1,l_2,\lambda_2} = c_{l_2,\lambda_2,l_1,\lambda_1}, \quad \forall l_1,\lambda_1,l_2,\lambda_2 \quad (6)$$

$$\sum_{l_1,\lambda_1} c_{l,\lambda,l_1,\lambda_1} \leq k, \quad \forall l,\lambda \quad (7)$$

$$p_{i,d} \in \{0,1\} \quad (8)$$

$$b_{i,d,l,\lambda} \in \{0,1\} \quad (9)$$

Constraint set (1) ensures that capacity is allocated for all the demands. Constraint set (2) ensures that primary-backup path pairs are used together for a demand. Constraint set (3) ensures that if any backup path uses a channel on a link, that channel is only allocated for backup.

- 5 Constraint set (4) ensures that demands sharing the backup channel are link disjoint in the primary. Constraint set (5) ensures that connections are setup between the channels when a backup path goes through them. Constraint set (6) ensures that connections in a K-shared switch are bidirectional. Constraint set (7) ensures that no more than K connections are made for each channel. Finally, constraint sets (8) and (9) enforce the integrality on the primary and backup
- 10 allocation.

Lastly, in step 606, these constraints and minimization criterion are submitted to an ILP solver to obtain the number of channels allocated to each link, and to obtain the paths and channels allocated for each demand. These results are considered the design of the K-shared network. The design may then be deployed.

5 In one illustrative embodiment, a K-shared network was designed using the methodologies described herein, which were written in AMPL language as described in R. Fourer, et al., "AMPL-A Modeling Language for Mathematical Programming," Boyd & Fraser, 1993, the disclosure of which is incorporated by reference herein, and solved using *CPLEX* ILP solver from ILOG, Inc.

10 IV. Illustrative Hardware Implementation

Referring now to FIG. 7, a block diagram illustrates a generalized hardware architecture of a computer system suitable for implementing an automated K-shared routing and design system, according to an embodiment of the present invention. More particularly, it is to be appreciated that any node in the network and/or any dedicated routing and/or design system in
15 the network may implement such a computing system 700 to perform the methodologies of the invention. Of course, it is to be understood that the invention is not limited to any particular computing system implementation.

In this illustrative implementation, a processor 702 for implementing at least a portion of the methodologies of the invention is operatively coupled to a memory 704, input/output (I/O)
20 device(s) 706 and a network interface 708 via a bus 710, or an alternative connection arrangement. It is to be appreciated that the term "processor" as used herein is intended to include any processing device, such as, for example, one that includes a central processing unit (CPU) and/or other processing circuitry (e.g., digital signal processor (DSP), microprocessor, etc.). Additionally, it is to be understood that the term "processor" may refer to more than one
25 processing device, and that various elements associated with a processing device may be shared by other processing devices.

The term "memory" as used herein is intended to include memory and other computer-readable media associated with a processor or CPU, such as, for example, random access

memory (RAM), read only memory (ROM), fixed storage media (e.g., hard drive), removable storage media (e.g., diskette), flash memory, etc.

In addition, the phrase "I/O devices" as used herein is intended to include one or more input devices (e.g., keyboard, mouse, etc.) for inputting data to the processing unit, as well as one
5 or more output devices (e.g., CRT display, etc.) for providing results associated with the processing unit. It is to be appreciated that such input devices may be one mechanism for a user to provide the design inputs used by a design system of the invention to generate a network. Alternatively, the design inputs could be read into the design system from a diskette or from some other source (e.g., another computer system) connected to the computer bus 710. Also,
10 inputs to the routing methodologies may be obtained in accordance with the one or more input devices. The output devices may be one mechanism for a user or other computer system to be presented with results of the routing and/or design methodologies.

Still further, the phrase "network interface" as used herein is intended to include, for example, one or more devices capable of allowing the computing system 700 to communicate
15 with other computing systems. Thus, the network interface may comprise a transceiver configured to communicate with a transceiver of another computer system via a suitable communications protocol. It is to be understood that the invention is not limited to any particular communications protocol.

It is to be appreciated that while the present invention has been described herein in the
20 context of routing and design systems, the methodologies of the present invention may be capable of being distributed in the form of computer readable media, and that the present invention may be implemented, and its advantages realized, regardless of the particular type of signal-bearing media actually used for distribution. The term "computer readable media" as used herein is intended to include recordable-type media, such as, for example, a floppy disk, a hard disk drive,
25 RAM, compact disk (CD) ROM, etc., and transmission-type media, such as digital or analog communication links, wired or wireless communication links using transmission forms, such as, for example, radio frequency and optical transmissions, etc. The computer readable media may take the form of coded formats that are decoded for use in a particular data processing system.

Accordingly, one or more computer programs, or software components thereof, including
30 instructions or code for performing the methodologies of the invention, as described herein, may be stored in one or more of the associated storage media (e.g., ROM, fixed or removable storage)

and, when ready to be utilized, loaded in whole or in part (e.g., into RAM) and executed by the processor 702.

In any case, it is to be appreciated that the techniques of the invention, described herein and shown in the appended figures, may be implemented in various forms of hardware, software, or combinations thereof, e.g., one or more operatively programmed general purpose digital computers with associated memory, implementation-specific integrated circuit(s), functional circuitry, etc. Given the techniques of the invention provided herein, one of ordinary skill in the art will be able to contemplate other implementations of the techniques of the invention.

Although illustrative embodiments of the present invention have been described herein with reference to the accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments, and that various other changes and modifications may be made by one skilled in the art without departing from the scope or spirit of the invention.